

A discussion of the EAI approach to hybrid computation

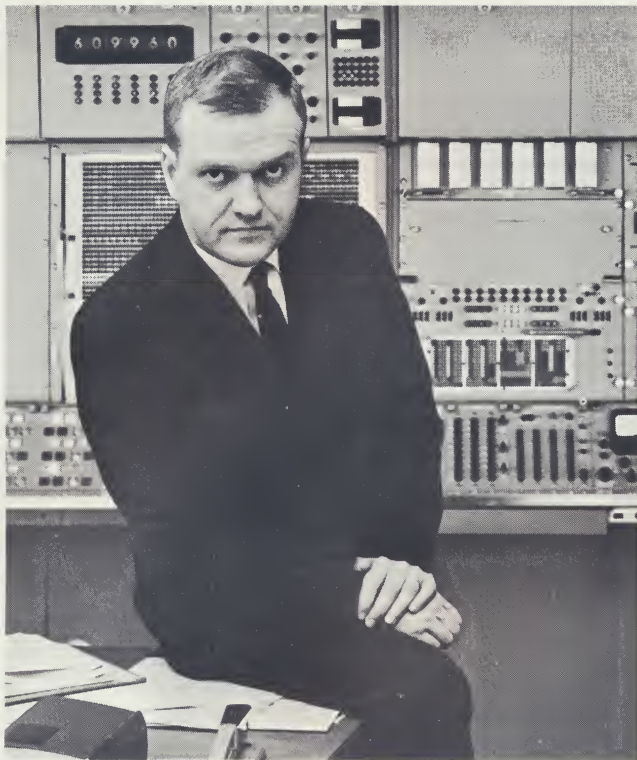
by T. D. TRUITT

*EAI Research and Computation Division
Princeton, New Jersey*



REPRINTED FROM THE OCTOBER 1965 ISSUE OF *SIMULATION*

Copyright© 1965 by Simulation Councils, Inc.



THOMAS D. TRUITT writes:

"I started out in Philadelphia, Pennsylvania. Later, 1946-48, I began to study electrical engineering at Swarthmore College, then paused for two years in the Navy—which stretched out to seven years—and I never found my way back to Swarthmore. As a pilot I enjoyed mapping Iceland and was bored with looking for submarines in the Atlantic Ocean. As an electronics officer in Texas, 1953-55, I was fascinated with the covey of operational flight trainers I tended. These analog computer devices infected me with simulationitis, which is difficult to cure. By parttime study and teaching at Texas A&I, I managed a BS in mathematics, and by hook or by crook left the Navy for Princeton University graduate school. Then I ran into GEDA and PACE computers and—of greatest fascination—the IAS (Institute of Advanced Study) digital computer. This interest in computers did more to divert my attention than it helped to earn the MSE degree.

"I joined EAI at the Princeton Computation Center in 1956, and benefitted immensely from a number of good analog studies. The next year I was engaged with the Datatron 205 at PCC. I wrote a digital simulation compiler called PACE (unpublished) to produce dynamic checks of analog programs. The 205 was shipped out before we could get serious about tying it to an analog computer. A strong interest in the DDA led to several years work on paper and breadboard designs of large "analog" complexes built from a kludge of digital circuits. Since 1960 hybrid and digital computer designs and programming have dominated my interests, but of course I don't do the work anymore—I am simply an idea collector and paper shuffler.

"I am now head of the Advanced Development Department in the Research and Computation Division of EAI, all of which means we study new computer designs, techniques, and programs—although currently 90% of the effort is directed toward software design and production.

"As a member of the IEEE and ACM, I still scan the tables of contents of their journals."

Mr. Truitt is also a member of SCi (and reads *SIMULATION*).

No. 7

A discussion of the EAI approach to hybrid computation

by T. D. TRUITT

EAI Research and Computation Division

Princeton, New Jersey

SUMMARY

The EAI 8900 Hybrid Computing System consists of an EAI 8800 analog computer, an EAI 8400 digital computer, associated analog and digital auxiliary equipment, and a simulation software system. This paper attempts to describe the 8900 system within the context of how and why computers are used for design simulation. From one point of view the hybrid computers available today fall neatly into two classes according to size. The small systems attempt only to achieve a useful degree of automation of repetitive runs on the analog computer or to increase its usefulness with digital function storage. The real impetus behind the expanding use of hybrid computers has been, of course, the need to simulate large, complex systems that defy economic or practical solution by other means. The very nature of these design studies then implies large analog and digital computers combined into an efficient hybrid simulation tool. The second and immediate implication of this is the need for comprehensive software to permit the simulation analyst to focus on his model rather than the computers.

A typical large-scale hybrid system is the EAI 8900, which combines the EAI 8800 and 8400 computers (analog and digital, respectively). The 8800 includes a large complement of 2-Mc logic elements and over 300 wide-bandwidth amplifiers. The 8400 is a large, fast arithmetic processor (32-bit word, up to 65,536 words of core storage, 2- μ second access, 6- μ second floating-point multiply). The 8400 has full control of the analog control system. The two systems communicate at high speed via 64 fully-buffered channels in both directions, at burst rates of 85,000 words per second A-to-D and 250,000 words per second D-to-A.

Simulation software

The 8900 software system program is quite different from that of a typical digital computer job shop, for it transforms the 8400 and 8800 computers into a single simulation design tool. The mode of operation at run-time, and hence the monitor, are unique; there is a yet unfilled need for a problem-oriented source language that is independent of the segmentation of the model equations between analog and digital computers. There are in use today several simulation (block-oriented) languages (MIDAS, MAD-BLOC, MIMIC) that generate digital codes for this type of problem. There is one program (APACHE) for generating analog patching instructions. The 8900 HYTRAN® programming system provides a single unified source language that performs both tasks after the programmer specifies the analog/digital segmentation. The same programming system will produce static and dynamic diagnostics for the hybrid program and will generate run-time control information for the monitor and for control and timing of the analog and data link equipment.

Model building is a process of putting together, in *parallel*, portions of a larger system. The above source language allows the programmer to think in terms of an effective *parallel* approach consistent with the model-building process rather than in terms of *sequential* processing of segments of code.

Design by computer simulation is ideally a process of *conversation* between the analyst and the computer model. The 8900 monitor system for a hybrid computer provides the analyst, during run-time, with a means for control, modification, and experimentation with his *model*—all in a symbolic form that is meaningful to him as a designer. Moreover, since the analyst must have time to think and to react, the monitor permits multiprogramming—keeping the 8400 active by time-sharing the on-line hybrid operations with the “background” job stack.

SIMULATION AND COMPUTERS

Change is the byword in scientific computers. Digital computers continue to become larger, smaller, more and less costly—and always faster. Analog computers are generally “larger” but in smaller packages, transistorized, “hybridized,”—and always faster. Whether due to revolution or evolution, it is evident that the time for needs and new ideas to yield new products in the scientific field is continually decreasing. Where is it all going? What claim does analog computation have on the future? In what form will hybrid computation reach maturity? What are the *real* computing needs in the simulation field? These are important questions to all members of SCi, vendors and users alike.

The key to the EAI answers, to the basic philosophy behind its new product line, lies within the very character of scientific simulation today. Simulation, a powerful and important step in the engineering design process, has grown in ten years from experimental studies of nonlinear control systems to very elaborate and mandatory investigations of complex engineering designs. The basic simulation procedures have been proven successful. Success has led to extrapolation to simulation of more complex systems. Heavier investments in new systems have made effective simulation more important and broader in scope. “A complex model carefully built up from verified models of subsystems may lead to valuable results attainable by no other means. Thus, as analysis and simulation have yielded understanding of the behavior of small systems, a natural process of escalation has led to simulation of systems of greater and greater complexity. The increased sophistication of simulation models has made the analyst even more dependent upon computers for effective control of the simulation and for interpretation of results.”¹ Finally, an increased tempo and economic factors demand more (and better) results from these studies per unit of time. Thus, two characteristics of simulation today are an *increase in complexity and scope* of the models and a *decrease in the time available* to obtain results.

If increased complexity and decreased time are important characteristics of today's simulation, then the implied computer requirements are for greater capacity, speed, flexibility, and automatic control. These are shopworn terms, but their increased significance is readily seen by consideration of the elements of the simulation process, as shown in figure 1. Greater analog and digital computer capacity is needed to handle the mathematical representation

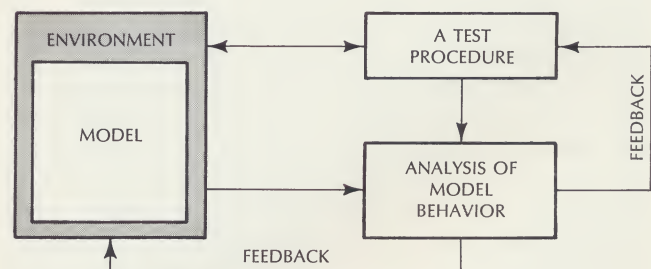


Figure 1—The elements of simulation in engineering design

A model and its environment are hypothesized; a test procedure is formulated to “operate” the model with various values of the model parameters and/or subject to various environmental conditions; the model behavior is analyzed and compared with theory or empirical data to create “feedback” decisions to change either the model, its environment, or the test procedure. The results include not only the behavioral data, but, more importantly, the total experience of the experimental design study.

of models of greater complexity as well as the synthesis of the model environments. The character of this capacity has changed in recent years. Models that must be studied over a wider dynamic range of behavior require wider-range performance from analog components, and in many cases numerical integration of some model variables. The study of fast, complex control systems has led to the inclusion of a significant amount of parallel logic in the analog computer. Function storage requirements have increased, leading to a greater demand for digital computation.

A computer model is not enough; a test procedure is needed. The analyst must design the experiment or investigation. He must predict the model behavior he expects, perform the experiment, and then compare and analyze the resultant behavior. After consideration, decisions are made to change the model parameters, the environmental conditions, or the test procedure. The steps are stated simply, but their effective implementation in many studies seriously taxes the best computer analyst. A really good simulation study is well planned long before the actual computer operation. The analyst knows, in general, the range of conditions he must test, the nature of the results to be expected, and the type of action he will take when evaluating intermediate results. In other words, good simulation procedures quite naturally lead to computer automation of much of the mechanics of the execution of a study. The degree of automation depends upon the size and nature of the study, but clearly a current computer requirement is for convenient automatic control of some portion of the elements in figure 1, and in particular, of those labeled "test procedure," "analysis of model behavior," and "feedback."

Speed of computation is almost always considered an important factor, since it is usually directly related to operating costs. There are two more important criteria which influence speed requirements: the need for real-time tie-in to external equipment (including man) and the importance of a fast response time between the analyst and the model, *i.e.*, the time constants of the feedback loops in figure 1. As the number of model parameters is increased, the number of conditions to be investigated — the dimensions of the study — must increase at least geometrically. Only faster computation speeds can bring the analyst the answers he needs in a short enough time. With these ideas in mind, the fundamental principles of the 8900 hybrid computer have been stated as:

- a. System software must be designed to fit the needs of the simulation analyst, *i.e.*, there must be effectual man-machine communication.

- b. A large analog computer requires not only speed and flexible control, but also a significant complement of parallel logic elements.
- c. The digital portion of a hybrid system must have the fastest possible arithmetic speeds commensurate with costs, and must have a very fast and flexible control interface system.
- d. The design of the data conversion devices and control signals between the parallel and sequential computer is crucial to efficient computation, and must permit the programmer a variety of modes of control.
- e. A large number of today's simulation problems do *not* require a full hybrid computer system; hence, its component parts must be separately useful for simulation work.

The EAI 8900 system (a combination of the 8400 and 8800 computers) is an outgrowth and an extension of earlier HYDAC systems. The EAI 8800 analog computer embodies most of the features of the HYDAC 2000. The EAI 8400 digital computer surpasses the processing and input/output of the digital computer portion of the HYDAC 2400. The EAI 8400 and 8800 computers are designed as a compatible pair to give an efficient hybrid computing system.

THE EAI 8800 ANALOG COMPUTER

The EAI 8800 is a wide-band analog computer, modularized for expansion from a small to a very large system. One dimension of expansion provides up to 70% of the logic computing power of the HYDAC 2000 systems. A second dimension adds a self-contained stored program and a digital I/O control system designed as an integral part of the 8800. Key features of the system are:

a. Capacity

The expanded 8800 console contains 60 combination integrator/summer amplifiers, 60 track/store summer amplifiers, and 228 additional operational amplifiers, all within the console proper. There are 240 servo-set attenuators. The nonlinear complement includes 48 quarter-square multipliers, 30 diode function generators, and six electronic resolvers which include 24 multipliers. The logic complement includes 72 flip-flops. Groups of four flip-flops alternatively act as binary (or BCD) counters or as shift registers. The main patch panel has 4080 holes for analog signals only, while a 2400-hole panel is used for logic elements and mode-control programming.

b. Speed

The computer has wide-band (200 Kc), solid-state, 100v amplifiers packaged directly behind the patch-panel to maintain wide-band system performance.

THE EAI 8400 DIGITAL COMPUTER

The EAI 8400 is a digital computer designed as a scientific processor in the medium price range with high arithmetic speed, interface flexibility, and programmer-oriented design features. Conventional software, as well as software specifically for hybrid simulation, complement the computer. A few of the most important features are:

a. Capacity

32-bit data word... memory expansion to 65,000 words... six sets of arithmetic instructions... sixteen Boolean connectives which operate on bytes of 1, 2, 4, 8, or 16 bits... over 750 instruction codes... and up to 272 priority interrupt lines.

b. Speed

A 2- μ second core memory, overlapped for faster execution at the 4K level... floating-point operations on 24-bit fraction and 8-bit exponent... floating add: 2.75-4.0 μ seconds... floating multiply: 6.0-7.5 μ seconds... floating divide: 8.75-10.0 μ seconds... double precision floating add: 3.5-6.0 μ seconds... 32-bit fixed-point add: 3.0-4.25 μ seconds. The extensive instruction repertoire results in fewer instructions being needed for each block of coding and thus in faster execution... a special "save" register for temporary storage of accumulator contents and a sixteen-word rapid access file have a 250-nanosecond access time, and are used to increase operating speeds.

c. Programming and operating aids

At the console, there is manual control of the auto load and auto dump features... four console interrupt buttons... a console register that may be used manually to control internal program parameters... the "universal accumulator" handles all operations for floating, fixed, and integer arithmetic, as well as mixed fixed and floating operations, without programmer attention to interregister operations.

Comments on the EAI 8400 design

To a programmer (such as the author), the 8400 appears as two machines merged into one. On the one hand, there is a powerful arithmetic processor, and, on the other, a flexible data processor and controller. An important dimension of a digital computer is its word size. The mathematical requirements in the simulation field dictate a fairly large arithmetic data word, while for scientific data processing and computer software applications the long word may be inefficiently employed. As a result, the 8400 employs 32 bits for most arithmetic operations, 56 bits for double-precision floating-point calculations, and 16 bits for nonarithmetic processing. In addition, there is direct addressing of all bytes of 8, 4, 2 or 1 bits.

The arithmetic power of the 8400 is best characterized by these three things:

- a. The computer was designed first and foremost for fast execution of floating-point instructions. The arithmetic section was optimized for speed, and the word size was chosen, in part, to permit use of one memory word per floating-point data word.
- b. There are six classes of arithmetic instructions: floating-point (32-bit), double-precision floating-point (56-bit), 32-bit fixed-point, 16-bit fixed-point, integer arithmetic (16-bit integers are automatically converted to floating-point numbers, permitting mixing of fixed and floating operations), and index arithmetic (16-bit operations on the seven index registers).
- c. The universal accumulator consists of several registers plus the arithmetic control circuitry. A programmer is never concerned with interregister transfers. The class of instruction automatically specifies the register configuration and the position of the data.

The experience of the EAI Research and Computation Division had indicated a precision requirement of 24 bits for data words for the majority of simulation problems. In all earlier computers, one paid a serious speed penalty for the use of floating-point arithmetic, so that fixed-point was standard for hybrid simulation work. The need for more speed, not less, was imperative for these problems in order to process any significant program in real time with the analog computer. We felt that if an economical floating-point arithmetic unit could be designed with execution times well under 10 μ seconds, it would yield a greater increase in overall processing speed than would a comparable investment in faster memory and circuits. This is so because of the saving of additional program steps required for fixed-point scaling. Scaling by shifting and multiplying is notably slow. Moreover, the programming convenience of automatic scaling was gained in addition to the 24-bit precision of the data word.

Similar considerations of ways of economically increasing overall speed led to the expansion of the instruction repertoire to a large number of instructions with numerous options. The net result is that one or two instructions will very often do the work of several, and do it in less time. At first, it may seem that a repertoire of 750 instructions would be cumbersome to work with. This, however, is not the case, for the natural grouping of instructions in classes with modifiers and convenient mnemonic codes leads to a set that is learned easily within a few hours.

The other face of the EAI 8400 is characterized by its flexibility as a processor of 16-bit words and

smaller bytes, by the convenience of 16-bit arithmetic in the accumulator and the index registers, and by the extensive provision for data and control signal exchange with external devices. All of these (including the 16 Boolean connectives) make for faster and more compact software programs and are expected to be most important in on-line control applications, such as in large flight simulators.

THE EAI 8900 HYBRID COMPUTER

The elements of the 8900 are illustrated in figure 2. Aside from the standard options and features of the two constituent computers, the dimensions of the system can be listed simply:

- Up to six 8800 computers attached to one 8400 computer
- Up to 32 conversion channels to and from each 8800
- Up to 64 conversion channels to and from the 8400

- Each 8800 logic panel terminates interrupts, sense lines, 8400-controlled flip-flops (in expandable quantities), and a 16-bit I/O data word.
- Analog-to-digital conversion burst rate is 84,000 words per second and digital-to-analog rate is 250,000 words per second.
- All A-to-D signals are buffered by separate track/store amplifiers, and all D-to-A transfers are buffered by two 15-bit registers followed by separate amplifiers with $\pm 100v$ output range.
- The track/store and DAC transfer operations may be synchronized by 8800 logic panel signals to achieve accurate timing for conversion operations.
- Block, sequential, and random address control of all conversion channels, as well as all other conversion functions, are completely under control of the 8400 program.
- All manual operational controls at the 8800 console are under direct control by the 8400 through the "control interface."

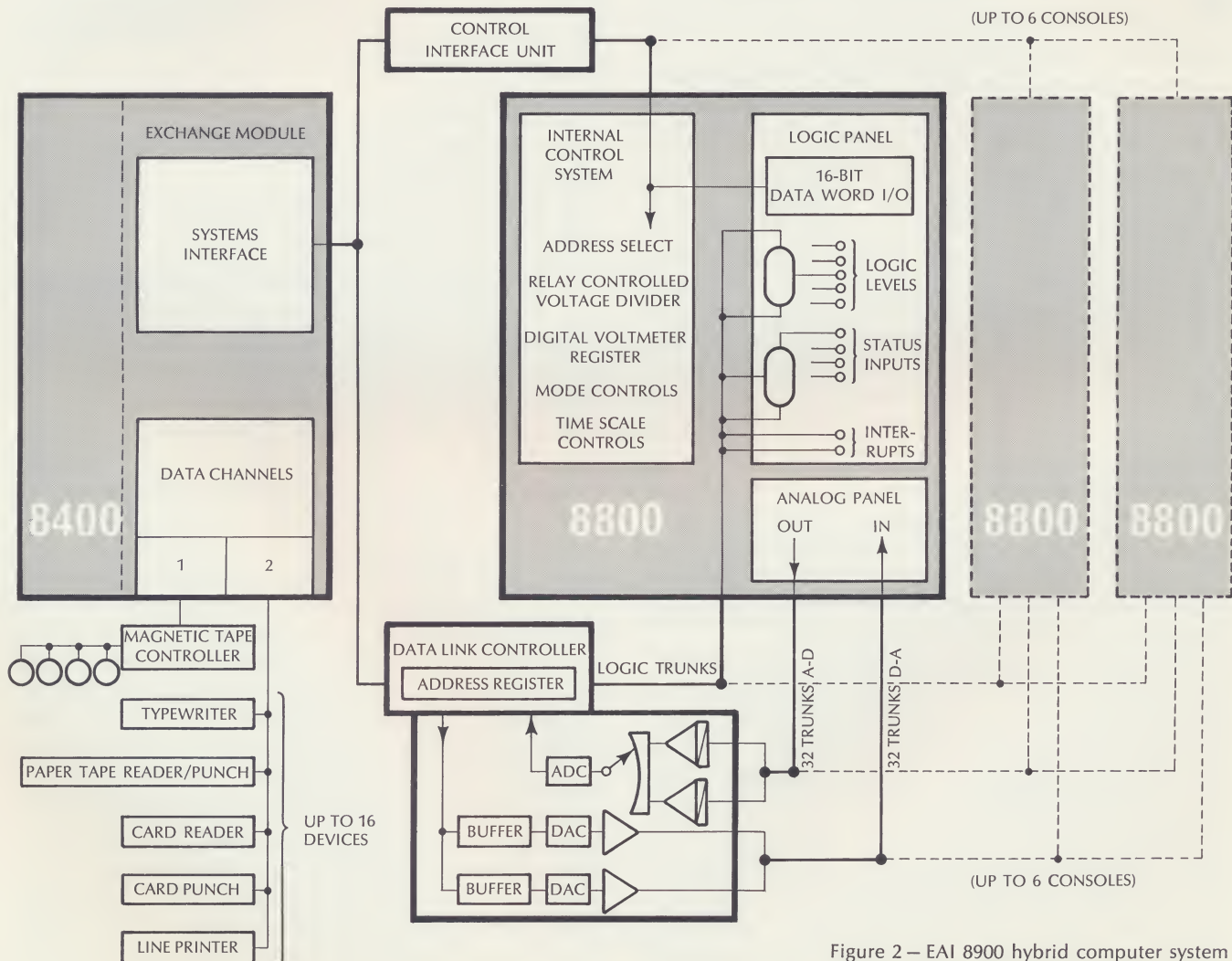


Figure 2 — EAI 8900 hybrid computer system

Simulation software

Large hybrid computing systems are complex. Programming experience on these systems is thin or nonexistent in most laboratories. It is not economical to spend long periods of machine time for idle programmers' experimentation. Rapid setup, checkout, and turn-around are imperative. Careful planning and documentation of the simulation model building and testing procedures, more than ever, are necessary for effective simulation studies. Programmer discipline in following standard procedures is a prerequisite for success. All these remarks lead to the not-so-surprising idea that comprehensive simulation software is as important to the simulation analyst and to the simulation laboratory manager as is the hybrid computer system—if not more so. Software is the bridge between the analyst and his computer-bound model. Software is the key to high machine utilization, to rapid setup and checkout, and to minimum programmer involvement with hardware details. The handmaiden of a software system is a system of standard programming procedures. The former fits the computer, the latter is for the user—so the two "speak" the same language.

The elements of the 8900 simulation software system are diagrammed in figure 3. The standard programming procedures that form a part of this system are indicated in figure 4, together with the sequence of events in program preparation for the 8900, which is discussed below. The software system may be grouped conveniently into utility programs, programming systems, and run-time control programs. Programming systems are the source-language processors or programs for preparation of machine-language code. These include a FORTRAN IV processor, a macroassembler, and HYTRAN programs for generation of both analog and digital programs and for static and dynamic check programs. The utility programs are smaller packages that are employed by the monitor or other programs for specific purposes. The run-time control systems are part of the simulation monitor. They provide the programmer on-line access to his simulation model and to all the controls of the analog computer.

The standard monitor is the master of the system. The resident executive is that part of the monitor that remains always in core storage. The executive supervises all I/O and interrupt operations and can call for any other programs it requires. The simulation monitor is an extension of the 8400 standard monitor to perform control of the entire 8900 system. The 8900 software system will perform on any configuration of 8400 and 8800 computers and peripheral equipment. All I/O and external control operations specify only functional units which must be assigned

to actual devices for a particular computer configuration. The assignment is accomplished by loading a table in memory with specific information on I/O devices and one or more 8800 analog computers. In the discussion below, the term "system file" refers to the file of all the systems programs shown in figure 3, which may be a file on magnetic tape, paper tape, or magnetic drum. Similarly, source and object programs may be loaded from card readers, paper-tape readers, or magnetic tapes (program file). The "job stack" is a file of programs to be batch processed, which may reside in the card-reader hopper, on drum, or on magnetic tape. The choice in each case is determined by the configuration table.

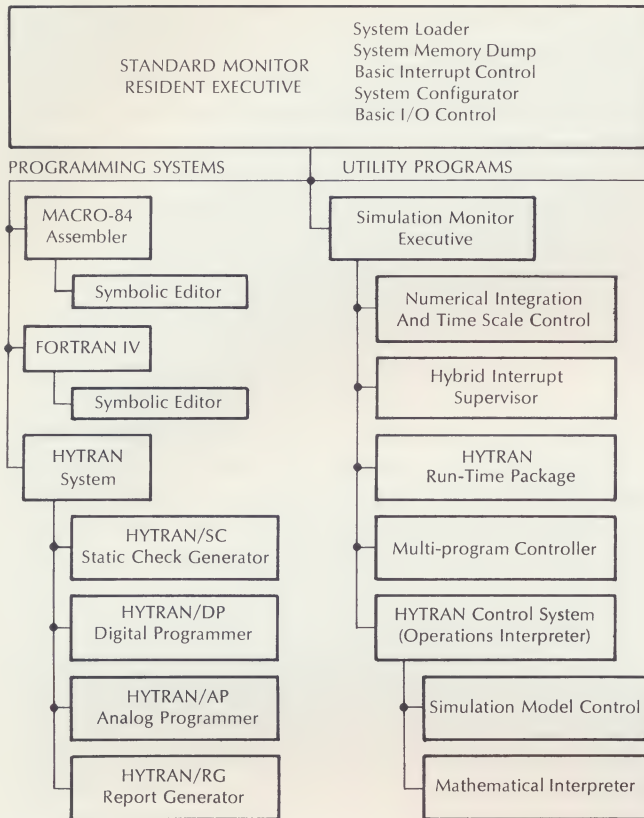
For the following discussion an 8900 configuration is assumed: 16K core, four magnetic tapes (systems file, program file, job stack file, and scratch file), card reader and punch, two line printers, and console typewriter. The card punch, one printer, and one magnetic tape drive are assigned to the job stack. The systems file, scratch file, card reader, and the 8400 processor are time-shared between the job stack and the 8900 simulation program under monitor control. The typewriter and the program file are reserved for the simulation program.

Program Preparation

The computer phase of a simulation design study requires three steps:

- a. Analysis and design of the model
- b. Program preparation and checkout
- c. Operation of and experimentation with the model

The first is a task for the analyst, the second employs the aid of the 8400, and the last uses the full 8900 system. In figure 4, these steps are shown in more detail. Discussion of procedures for proper formulation of a mathematical model is a treatise in itself and is not covered here. Procedures for diagramming a simulation model are part of the 8900 systems manuals and follow, in principle, the experience of analog computer simulation work. Note here that the entire model is to be mapped out and that then an allocation of segments of the model is made between the 8800 and 8400 computers, based on a set of rules developed from hybrid computing experience. Segments of the model assigned to the 8400 are divided between calculations that are time-base-dependent (e.g., integration of differential equations in time), and those that are not, such as arbitrary function generation and algebraic calculations. Each of these is subdivided according to one or more ranges of real-time time-constants (*i.e.*, maximum expected frequency or rate of change of the model variables with respect to a real clock). Each subdivi-



sion is prepared as if an independent program. The timing and linkage of these subprograms is accomplished by the control program and the simulation monitor.

Figure 3 — EAI 8900 simulation software system

The process of allocation of the model between the parallel and sequential programs is only one step in the analysis of the model. As in all types of programming, the computer forces the analyst to ask certain questions about the nature of his model. In this case the questions have to do with the relative magnitude of system time-constants and the influ-

SF — SYSTEM FILE
PF — PROGRAM FILE

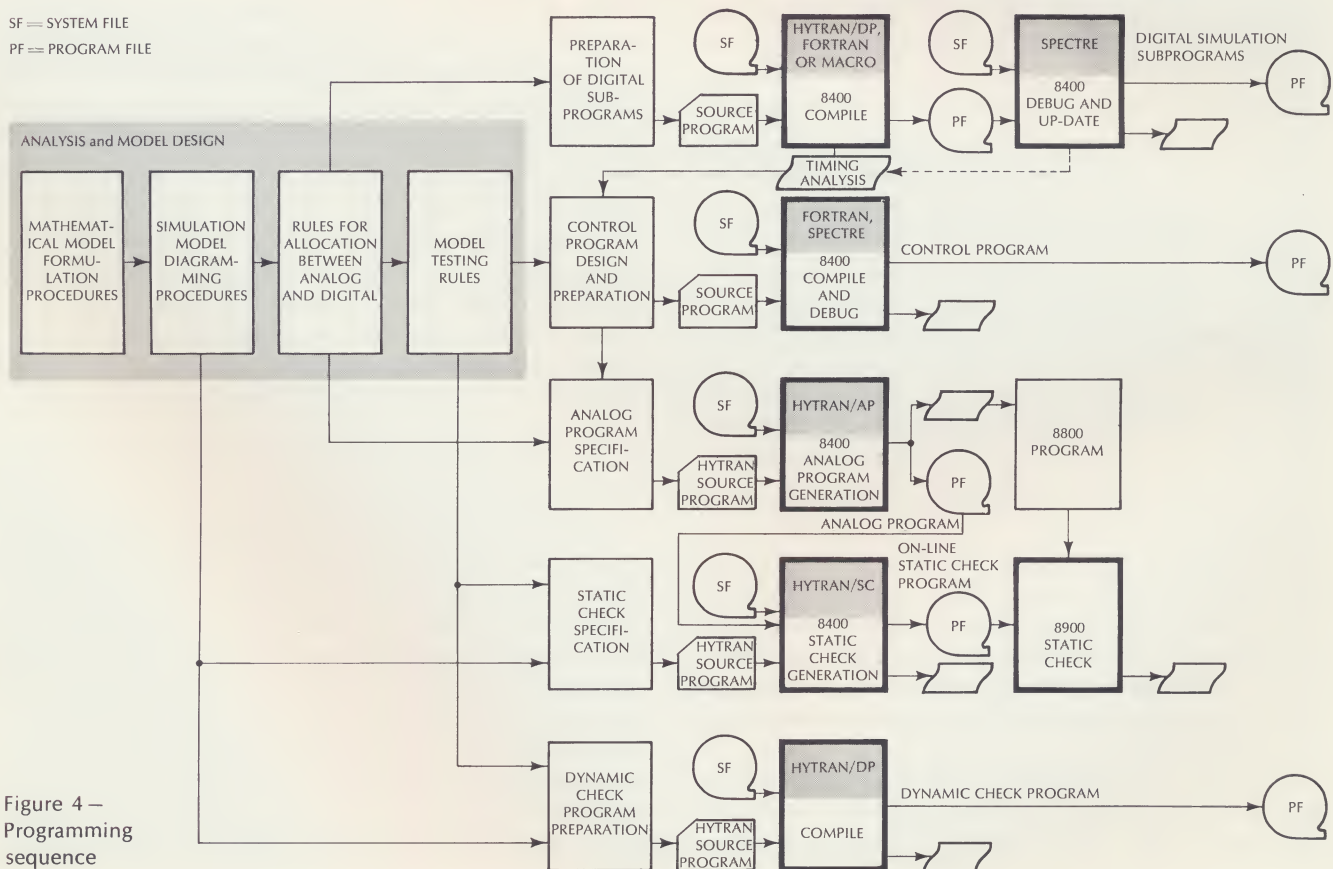


Figure 4 — Programming sequence

ence upon system performance. The experienced analyst can always make estimates for the answers to these questions. The next step in the study is to determine a method of testing the performance of the simulation model (figure 1). This may be the most crucial step in the entire study. The analyst must determine, at this time, exactly what experiments he wishes to make and what performance he expects to see, and then, if failure occurs, what he wishes to do about it: change parameters, change the model, or change the test procedure. With a thorough guide book of rules the analyst is led through a list of the many things he might care to try, and he must then select and specify each step of the simulation study. This information forms the basis for the control program design.

Referring again to figure 4, the program preparation phase is seen to follow the first four analytical steps on the left, and proceed by five parallel paths from left to right. The sequence of events is indicated by the relative position of the blocks. As shown on the lowest path, once the simulation model has been diagrammed and a set of test conditions selected, a dynamic check program can be generated for later evaluation. HYTRAN/DP, a processor called "Digital Programmer," translates the diagrammatic HYTRAN source language description of the entire model into an 8400 program that will calculate the dynamic performance of the model for one or two selected test cases. When this program is executed the results will be compared with the solution of the same test cases performed by the 8900 simulation. The dynamic check program is not executed until the static check is performed and the suitability of test case parameters and the analog scaling has been verified.

The second path from the bottom of figure 4 indicates the preparation of the HYTRAN static check program. This check procedure consists of two steps: (a) the preliminary off-line check, performed by the 8400 where the proposed analog program is verified against the original equations or mathematical model specification, and (b) the on-line check where the 8900 executes a program generated in step (a) to verify the actual state of the 8800 (with patch panel in place) as well as the state of the 8400 simulation variables. This on-line check program can be called and executed at any time during the running of the simulation model, for the program is independent of the original test conditions. With the 8900 in HOLD mode, the program reads the values of all state and control variables and employs them as the conditions for the test.

The 8900 object program consists of three pieces: digital subprograms, a control program (stored on the program file), and an 8800 program prepared on

the analog and logic patch panels. The digital subprograms are prepared by one of three programming systems (HYTRAN/DP, FORTRAN IV, MACRO-84) according to the choice of the programmer. In fact, the use of the three may be mixed. The compilation, or assembly, of these programs produces estimates of the execution times for each subprogram, and even for each statement. This information is important in the design of the control program—even rough estimates are valuable and hence may be used—prior to final checkout of the subprograms. Source and unchecked object programs are placed on the program file, the SPECTRE debug system is called by the monitor, and the programmer can proceed to debug the subprograms at a source statement or at symbolic assembly language level, using the console typewriter as if the whole machine were his—the SPECTRE system will interrupt the job stack programs to time-share the machine with the debugging commands from the typewriter.

The control program is prepared and checked out as above and placed with the digital subprograms on the program file. The control program is primarily an implementation of a sequence of model test, evaluations, and decisions. This program will most often be prepared in FORTRAN IV source language, although assembly coding may, in special cases, be mixed in-line with FORTRAN statements.

The preparation of the detailed analog and logic control programs currently is accomplished by the programmer. Since this programming process can be described as a rational one, its automation by means of an 8400 program (HYTRAN/Analog Programmer) is obviously a desirable step. The objective of this program is similar to that of the APACHE system,² but is to encompass hybrid as well as analog programs. Moreover, a common source language is employed for HYTRAN/AP and other HYTRAN processors—a simulation language, block oriented yet with a statement syntax, of the genre of MADBLOC, MIDAS, and DSL-90.³

Running the hybrid simulation

With the completion of all the steps in figure 4, the analyst has a computer model that has been checked out for at least one static condition. This means that all algebraic "loops" in his model have been checked, but that the time-dependent behavior must yet be checked. It is time now for the control program to exercise subsystems and to check for expected dynamic behavior. The control program is planned to run through several such simple cases, and during run time the analyst may care to specify a few additional dynamic tests before calling for execution of

the full-scale dynamic test comparison of 8900 results with the completely digital check calculations.

Several steps must now be taken to prepare for running the simulation model. It has been assumed so far that the 8400 is operating as a batch processor (except when on-line debugging by SPECTRE was mentioned above), but now the simulation job must be given priority over the job stack whenever it requires it. Whenever the simulation is stopped, even momentarily, for human intervention, evaluation of results, or for control of slow operations on the analog computer (pot setting), the 8400 will return to process the work in the job stack until called for simulation duty again. The procedure to prepare for running the model is outlined in figure 5.

The action of the first seven boxes in figure 5 is initiated by control cards (or typewriter) and the set-up of the system proceeds automatically. When setup is complete the multiprogram controller transfers to the job stack while the programmer decides what steps he will take next. From the typewriter he can initiate a dynamic check comparison or static check operation (HYTRAN run-time package), or a modification of the 8800 (HYTRAN control system) as described above.

When all preliminary check procedures are complete, the programmer transfers control, via typewriter, to the control program. This program proceeds in a systematic fashion to put the model through a predetermined set of runs with varying parameters, initial conditions, or driving functions. Results are displayed upon analog recording devices, and some may be digitized and stored on the scratch file. The control program keeps the analyst informed about the progress of the test program. At any point he may hold up the test sequence, specify new tests, modify the analog or digital portion of model structure, and request additional runs and/or different output recordings.

An element which is essential to the process of engineering design by simulation is the "conversational" nature of the interaction between the analyst and his simulation model. This process of synthesis is an experimental one—a planned procedure of trial, measurement, adjustment, and retrial with the analyst in the decision loop to evaluate the unexpected and to employ some degree of intuition to the choice of subsequent experiments. All of this requires software tools that are quite different from those found in a typical scientific digital computer laboratory. Simulation must bring the programmer to the computer console with tools that permit direct conversation with his simulation model—but at the same time maintaining efficient use of the computer system. The 8900 software system is geared to provide this cru-

cial mode of operating. The primary facilities employed to accomplish this are (a) the multiprogram control of the simulation monitor, (b) the use of fully "reentrant" or interruptable programs together with the priority interrupt system of the 8400, (c) dynamic storage reallocation control for maximum utilization of memory, and (d) the syntax-directed HYTRAN on-line control language.

1. Place patch panels on 8800(s).
2. Bypass 8400 job stack via typewriter command that calls simulation monitor from system file.
3. Load control cards for the following 8400 sequence.

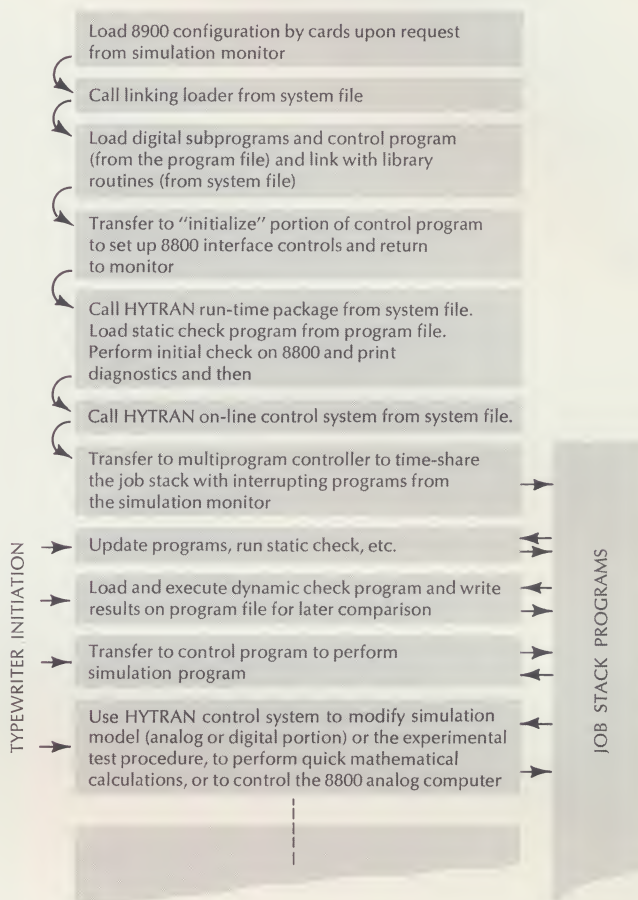


Figure 5 — EAI 8900 run-time procedures

REFERENCES

- 1 T D TRUITT
Hybrid computation . . . what is it? . . . who needs it?
AFIPS Conference Proceedings vol 25 pp 249-269
- 2 C GREEN A DEBROUX G P del BIGIO H D'HOOP
APACHE—A breakthrough in analog computing
IRE Transactions on Electronic Computers
October 1962 pp 699-706
- 3 cf: *SIMULATION* March 1965 pp 142-143